EX1:

a. Use EM algorithms and the generated samples to estimate the unknown parameters

m = [1 1; 5 5; 9 1]'

S = [1 0.4; 0.4 1;

1 -0.6; -0.6 1;

1 0; 0 1];

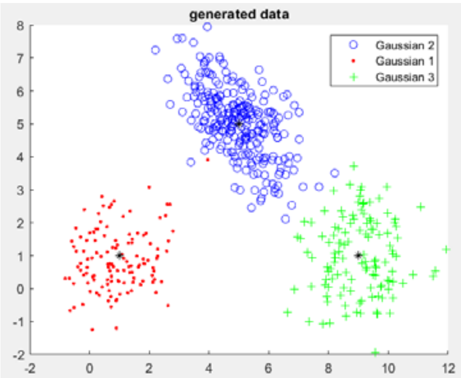
Generate 500 samples according "the first two samples are generated from the 2nd Gaussian, the 3rd sample from the 1st one, and the 4th sample from the last Gaussian".

X

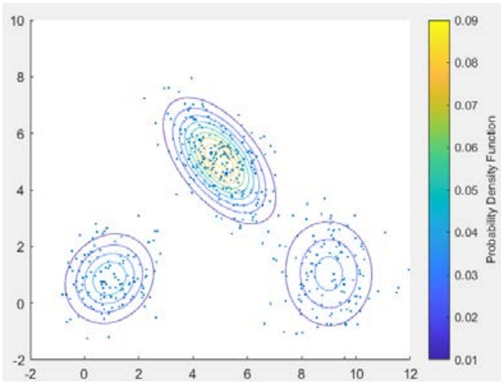
2x500 double

	1	2	3	4	5	6	7	8	9	10
1	6.1650	5.0751	0.3035	9.0591	5.2641	3.5538	2.2460	9.5774	4.8644	3.7296
2	4.8025	5.2362	2.2759	2.7971	5.5389	5.3068	0.9128	0.6400	4.0019	6.5499

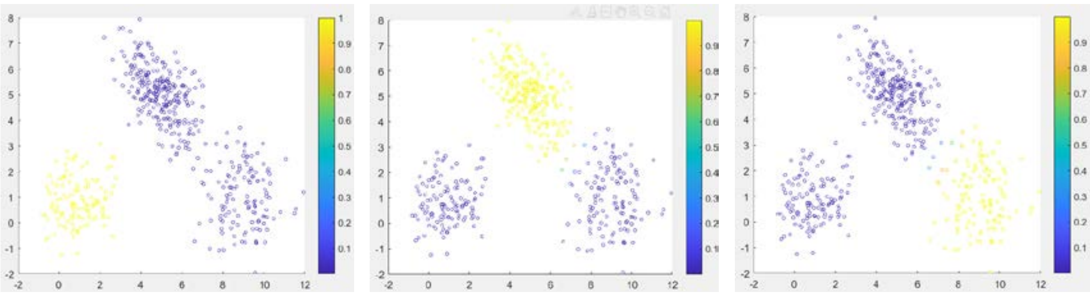
Generate ground truth data



Probability Density Function



Posterior Probability of Component 1, 2, 3



Result:

P1 = 0.25016

P2 = 0.49691

P3 = 0.25293

```
mu1 =
    0.9428    0.9157

mu2 =
    4.9957    5.0086

mu3 =
    8.9792    0.8659
```

```
sigma1 =
    0.9773    0.2663
    0.2663    1.0275

sigma2 =
    1.0868   -0.5956
   -0.5956    0.9393

sigma3 =
    0.7765    0.1923
    0.1923    0.8832
```

b. Same as above

m = [1 1; 3.5 3.5; 6 1]'

S = [1 0.4; 0.4 1;

1 -0.6; -0.6 1;

1 0; 0 1];

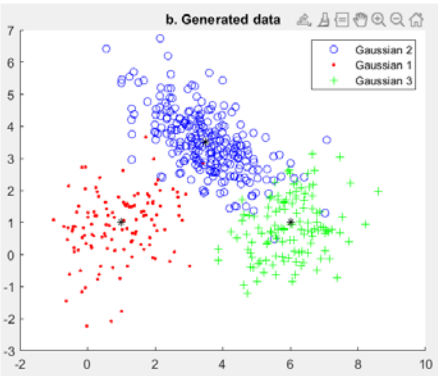
Generate 500 samples according "the first two samples are generated from the 2nd Gaussian, the 3rd sample from the 1st one, and the 4th sample from the last Gaussian".

X

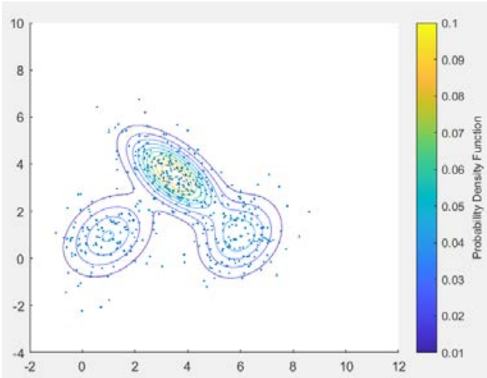
2x500 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	4.0377	1.2412	1.3188	5.5664	7.0784	2.1501	1.7254	6.7147	3.3759	4.9090	1.6715	6.7172	3.9
2	4.6445	5.5450	-0.0710	1.3426	3.5685	6.7379	1.2324	0.7950	4.7662	3.7883	0.1619	2.6302	4.0

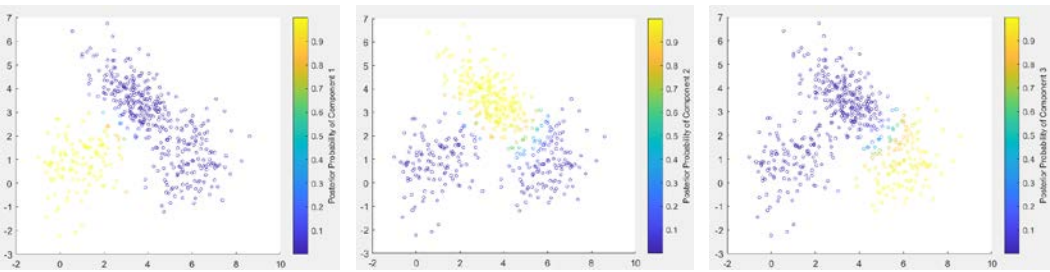
Generate ground truth data



Probability Density Function



Posterior Probability of Component 1, 2, 3



Result:

P1 = 0.25446

P2 = 0.48436

P3 = 0.26117

```
mu1 =
      0.9644      0.9475

sigma1 =
      1.0013      0.2997
      0.2997      1.0587

mu2 =
      3.4751      3.5171

sigma2 =
      1.0113     -0.6312
     -0.6312      0.9790

mu3 =
      5.9760      0.9276

sigma3 =
      0.8011      0.1935
      0.1935      0.9825
```

c. Same as above

m = [1 1; 3.5 3.5; 6 1]'

S = [1 0.4; 0.4 1;

1 -0.6; -0.6 1;

1 0; 0 1];

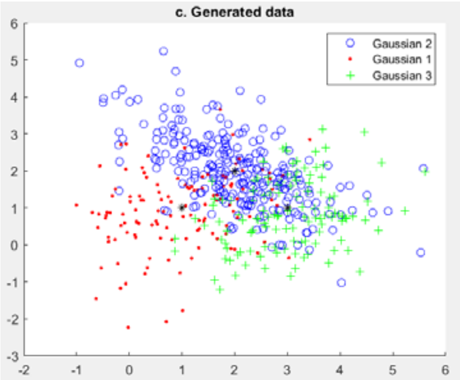
Generate 500 samples according "the first two samples are generated from the 2nd Gaussian, the 3rd sample from the 1st one, and the 4th sample from the last Gaussian".

X

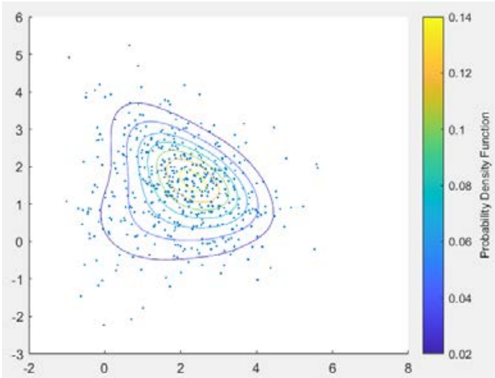
2x500 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2.5377	-0.2588	1.3188	2.5664	5.5784	0.6501	1.7254	3.7147	1.8759	3.4090	1.6715	3.7172	2.4
2	3.1445	4.0450	-0.0710	1.3426	2.0685	5.2379	1.2324	0.7950	3.2662	2.2883	0.1619	2.6302	2.5

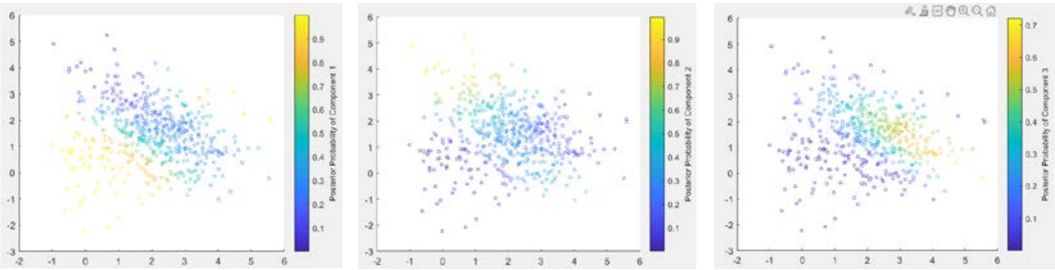
Generate ground truth data



Probability Density Function



Posterior Probability of Component 1, 2, 3



Result:

P1 = 0.44565

P2 = 0.2924

P3 = 0.26195

```
mu1 =
    1.7474    0.9900

mu2 =
    1.7912    1.9556

mu3 =
    2.6222    1.6183
```

```
sigma1 =
    1.7980    0.2777
    0.2777    1.1010

sigma2 =
    1.1219   -0.8495
   -0.8495    1.5098

sigma3 =
    0.9813   -0.4760
   -0.4760    0.5891
```

In case of C. It's failed to converge in 100 iterations, which I have setting, so got the worse result.

d.

(2D-visiualize as above.)

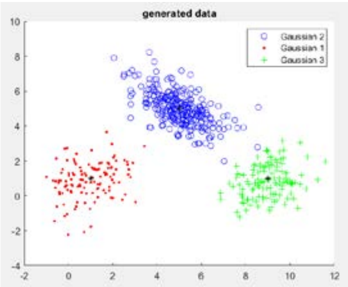
The original data set, which consists of three overlapping groups of points.

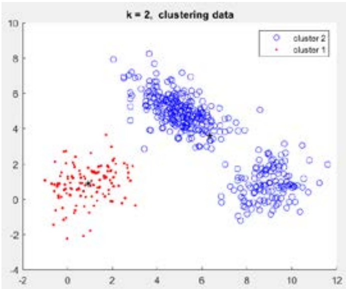
Trying to use Gaussian mixtures to get a more obvious clustering. Each one of the obtained clusters contains a significant percentage of points from more than one distribution.

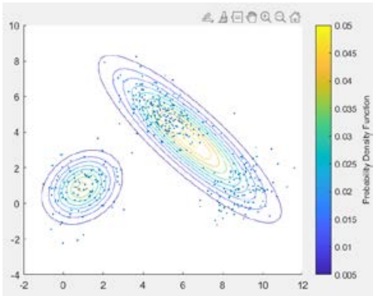
EX2 K-means for

(a) k=2

Generate ground truth data | k=2 clustering data | Probability Density Function





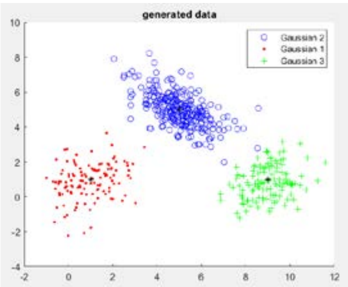


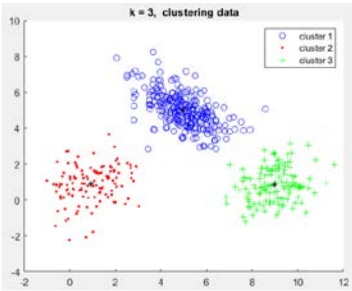
Result:

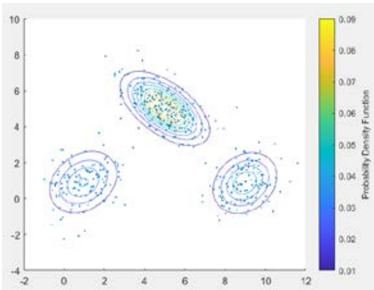
```
P1 = 0.24971
P2 = 0.75029

mu1 =
    0.9214
    0.8977

mu2 =
    6.3309
    3.6094

sigma1 =
    0.9686    0.2575
    0.2575    1.0202

sigma2 =
    4.5320   -4.0155
   -4.0155    4.7542
```

(b) k=3

Generate ground truth data | k=3 clustering data | Probability Density Function





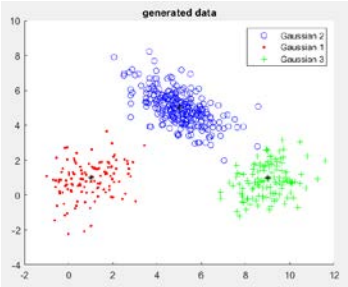


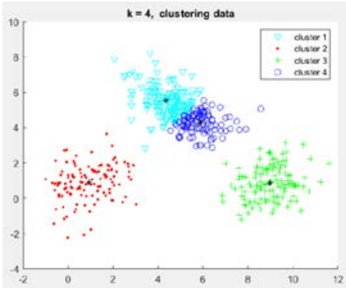
Result:

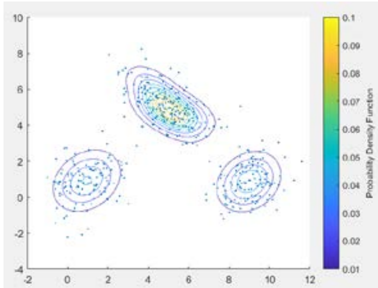
```
P1 = 0.25295
P2 = 0.49689
P3 = 0.25016

mu1 =
    0.9214
    0.8977

mu2 =
    4.9839
    5.0073

mu3 =
    8.9720
    0.8688

sigma1 =
    0.7765    0.1921
    0.1921    0.8838

sigma2 =
    1.0865   -0.5954
   -0.5954    0.9392

sigma3 =
    0.9773    0.2663
    0.2663    1.0275
```

(c) k=4

Generate ground truth data　　k=4 clustering data　　Probability Density Function







Result:

P1 = 0.29539
P2 = 0.25319
P3 = 0.2013
P4 = 0.25012

```
mu1 =
    0.9214
    0.8977

mu2 =
    5.8878
    4.2440

mu3 =
    8.9720
    0.8688

mu4 =
    4.3670
    5.5282
```

```
sigma1 =
    0.9136   -0.6623
   -0.6623    1.1224

sigma2 =
    0.7789    0.1909
    0.1909    0.8754

sigma3 =
    1.1122   -0.2694
   -0.2694    0.4512

sigma4 =
    0.9768    0.2651
    0.2651    1.0255
```

EX3 Fuzzy k-means algorithm (k=3)

(a) q=2

U

3x500 double

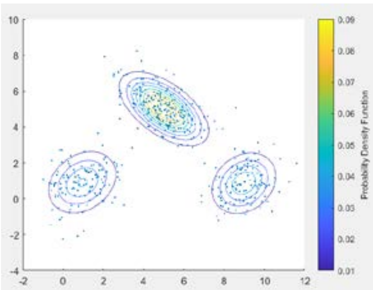
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.0304	0.1613	0.9536	0.0052	0.0899	0.1447	0.9550	0.0078	0.0318	0.0402	0.9473	0.0374	0.0
2	0.9312	0.7514	0.0279	0.0114	0.5141	0.7465	0.0305	0.0149	0.9367	0.8815	0.0321	0.1052	0.9
3	0.0383	0.0872	0.0185	0.9834	0.3960	0.1088	0.0145	0.9773	0.0314	0.0783	0.0207	0.8574	0.0

Confusion matrix:

P1 = 0.25295
P2 = 0.49689
P3 = 0.25016

```
mu1 =
    0.9214
    0.8977

mu2 =
    4.9839
    5.0073

mu3 =
    8.9720
    0.8688
```

```
sigma1 =
    0.7765    0.1921
    0.1921    0.8838

sigma2 =
    1.0865   -0.5954
   -0.5954    0.9392

sigma3 =
    0.9773    0.2663
    0.2663    1.0275
```



(b)q=3

U

3x500 double

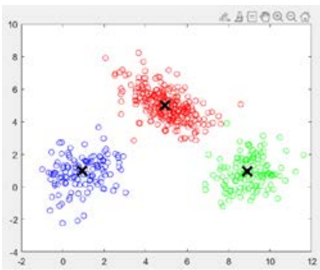
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.1330	0.2591	0.7543	0.0578	0.1825	0.2436	0.7757	0.0768	0.1374	0.1424	0.7453	0.1338	0.0
2	0.7181	0.5513	0.1354	0.0855	0.4337	0.5459	0.1327	0.1058	0.7264	0.6588	0.1413	0.2238	0.7
3	0.1490	0.1896	0.1103	0.8567	0.3838	0.2105	0.0916	0.8175	0.1362	0.1988	0.1135	0.6423	0.1

Confusion matrix:

P1 = 0.248
P2 = 0.496
P3 = 0.256

```
mu1 =
    0.9179    0.9738

mu2 =
    4.9240    5.0168

mu3 =
    8.9034    0.9514
```

```
sigma1 =
    0.7765    0.1921
    0.1921    0.8838

sigma2 =
    1.0865   -0.5954
   -0.5954    0.9392

sigma3 =
    0.9773    0.2663
    0.2663    1.0275
```



FCM 的群心是藉由 "所有的數據點" 以 "距離乘上隸屬值" 而成。其中 q 值稱為 fuzzier ，就是 FCM 中的 Fuzzy ，此值可以調控該項對於群心計算的影響力。